Deep Learning for Single Image Super-Resolution: A Brief Review

Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, Qingmin Liao

***Abstract*—Single image super-resolution (SISR) is a notoriously challenging ill-posed problem that aims to obtain a high-resolution (HR) output from one of its low-resolution (LR) versions. Recently, powerful deep learning algorithms have been applied to SISR and have achieved state-of-the-art performance. In this survey, we review representative deep learning-based SISR methods and group them into two categories according to their contributions to two essential aspects of SISR: the exploration of efficient neural network architectures for SISR and the development of effective optimization objectives for deep SISR learning. For each category, a baseline is first established, and several critical limitations of the baseline are summarized. Then, representative works on overcoming these limitations are presented based on their original content, as well as our critical exposition and analyses, and relevant comparisons are conducted from a variety of perspectives. Finally, we conclude this review with some current challenges and future trends in SISR that leverage deep learning algorithms.**

***Index Terms*—Single image super-resolution, deep learning, neural networks, objective function**

I. Introduction

DEEP learning (DL) [1] is a branch of machine learning algorithms that aims at learning the hierarchical representations of data. Deep learning has shown prominent superiority over other machine learning algorithms in many artificial intelligence domains, such as computer vision [2], speech recognition [3], and natural language processing [4]. Generally, the strong capacity of DL to address substantial unstructured data is attributable to two main contributors: the development of efficient computing hardware and the advancement of sophisticated algorithms.

Single image super-resolution (SISR) is a notoriously challenging ill-posed problem because a specific low-resolution (LR) input can correspond to a crop of possible high-resolution (HR) images, and the HR space (in most instances, it refers to the natural image space) that we intend to map the LR input to is usually intractable [5]. Previous methods for SISR mainly have two drawbacks: one is the unclear definition of the mapping that we aim to develop between the LR space and the HR space, and the other is the inefficiency of establishing a complex high-dimensional mapping given massive raw data. Benefiting from the strong capacity of extracting effective high-level abstractions that bridge the LR and HR space, recent DL-based SISR methods have achieved significant improvements, both quantitatively and qualitatively.

In this survey, we attempt to give an overall review of recent DL-based SISR algorithms. We mainly focus on two areas: efficient neural network architectures designed for SISR and effective optimization objectives for DL-based SISR learning. The reason for this taxonomy is that when we apply DL algorithms to tackle a specified task, it is best for us to consider both the universal DL strategies and the specific domain knowledge. From the perspective of DL, although many other techniques such as data preprocessing [6] and model training techniques are also quite important [7], [8], the combination of DL and domain knowledge in SISR is usually the key to success and is often reflected in the innovations of neural network architectures and optimization objectives for SISR. In each of these two focused areas, based on the benchmark, several representative works are discussed mainly from the perspective of their contributions and experimental results as well as our comments and views.

The rest of the paper is arranged as follows. In Section II, we present relevant background concepts of SISR and DL. In Section III, we survey the literature on exploring efficient neural network architectures for various SISR tasks. In Section IV, we survey the studies on proposing effective objective functions for different purposes. In Section V, we summarize some trends and challenges for DL-based SISR. We conclude this survey in Section VI.

II. Background

A. Single Image Super-Resolution

Super-resolution (SR) [9] refers to the task of restoring high-resolution images from one or more low-resolution observations of the same scene. According to the number of input LR images, the SR can be classified into single image super-resolution (SISR) and multi-image super-resolution (MISR). Compared with MISR, SISR is much more popular because of its high efficiency. Since an HR image with high perceptual quality has more valuable details, it is widely used in many areas, such as medical imaging, satellite imaging and security imaging. In the typical SISR framework, as depicted in Fig. 1, the LR image y is modeled as follows:

$$y = (x \otimes k)\downarrow_s + n, \tag{1}$$

This work was partly supported by the National Natural Science Foundation of China (Nos.61471216 and 61771276), the National Key Research and Development Program of China (No.2016YFB0101001) and the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen (Nos. JCYJ20170307153940960 and JCYJ20170817161845824). (Corresponding author: Wenming Yang)

W. Yang, X. Zhang, W. Wang and Q. Liao are with the Department of Electronic Engineering, Graduate School at Shenzhen, Tsinghua University, China (E-mail: {yang.wenming@sz, xc-zhang16@mails, wangwei17@mails, liaoqm@}.tsinghua.edu.cn.

Y. Tian is with the University of Rochester, USA (E-mail: ytian21@ur.rochester.edu).

J.-H. Xue is with the Department of Statistical Science, University College London, UK (E-mail: jinghao.xue@ucl.ac.uk).

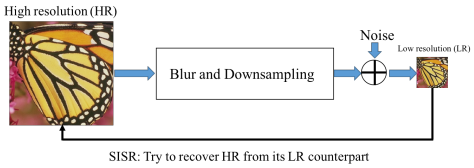


Figure 1: Sketch of the overall framework of SISR.

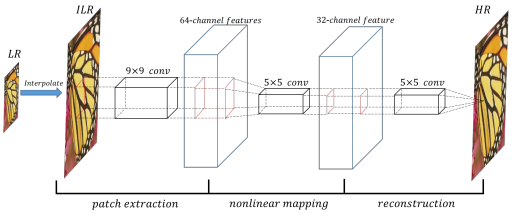


Figure 2: Sketch of the SRCNN architecture.

where $x \otimes k$ is the convolution between the blurry kernel k and the unknown HR image x, $\downarrow_s$ is the downsampling operator with scale factor s, and n is the independent noise term. Solving (1) is an extremely ill-posed problem because one LR input may correspond to many possible HR solutions. To date, mainstream algorithms of SISR are mainly divided into three categories: interpolation-based methods, reconstruction-based methods and learning-based methods.

Interpolation-based SISR methods, such as bicubic interpolation [10] and Lanczos resampling [11], are very speedy and straightforward but suffer from accuracy shortcomings. Reconstruction-based SR methods [12], [13], [14], [15] often adopt sophisticated prior knowledge to restrict the possible solution space with an advantage of generating flexible and sharp details. However, the performance of many reconstruction-based methods degrades rapidly when the scale factor increases, and these methods are usually time-consuming.

Learning-based SISR methods, also known as example-based methods, are brought into focus because of their fast computation and outstanding performance. These methods usually utilize machine learning algorithms to analyze statistical relationships between the LR and its corresponding HR counterpart from substantial training examples. The Markov random field (MRF) [16] approach was first adopted by Freeman *et al.* to exploit the abundant real-world images to synthesize visually pleasing image textures. Neighbor embedding methods [17] proposed by Chang *et al.* took advantage of similar local geometry between LR and HR to restore HR image patches. Inspired by the sparse signal recovery theory [18], researchers applied sparse coding methods [19], [20], [21], [22], [23], [24] to SISR problems. Lately, random forest [25] has also been used to achieve improvement in the reconstruction performance. Meanwhile, many works combined the merits of reconstruction-based methods with the learning-based approaches to further reduce artifacts introduced by external training examples [26], [27], [28], [29]. Very recently, DL-based SISR algorithms have demonstrated great superiority to reconstruction-based and other learning-based methods.

B. Deep Learning

Deep learning is a branch of machine learning algorithms based on directly learning diverse representations of data [30]. In contrast to traditional task-specific learning algorithms that select useful handcrafted features with expert domain knowledge, deep learning algorithms aim to learn informative hierarchical representations automatically and then leverage them to achieve the final purpose, where the whole learning process can be seen as an entirety [31].

Because of the high approximating capacity and hierarchical property of an artificial neural network (ANN), most modern deep learning models are based on ANNs [32]. Early ANNs can be traced back to perceptron algorithms in the 1960s [33]. Then, in the 1980s, the multilayer perceptron could be trained with the backpropagation algorithm [34], and the convolutional neural network (CNN) [35] and recurrent neural network (RNN) [36], two representative derivatives of the traditional ANN, were introduced to the computer vision and speech recognition fields, respectively. Despite remarkable progress achieved by ANNs during that period, there were still many deficiencies handicapping ANNs from developing further [37], [38]. Thereafter, the rebirth of the modern ANN was marked by pretraining the deep neural network (DNN) with the restricted Boltzmann machine (RBM) proposed by Hinton in 2006 [39]. Consequently, benefiting from the boom of computing power and the development of advanced algorithms, models based on the DNN have achieved remarkable performance in various supervised tasks [40], [41], [2]. Meanwhile, DNN-based unsupervised algorithms such as the deep Boltzmann machine (DBM) [42], variational autoencoder (VAE) [43], [44] and generative adversarial nets (GAN) [45] have attracted much attention owing to their potential to address challenging unlabeled data. Readers can refer to [46] for an extensive analysis of DL.

III. Deep Architectures for SISR

In this section, we mainly discuss the efficient architectures proposed for SISR in recent years. First, we set the network architecture of super-resolution CNN (SRCNN) [47], [48] as the benchmark. When we discuss each related architecture in detail, we focus on their universal parts that can apply to other tasks and their specific parts that characterize SISR properties. To meaningfully construct fair comparisons among different models, we will illustrate the importance of the training dataset and attempt to compare models with the same training dataset.

A. Benchmark of Deep Architecture for SISR

We select the SRCNN architecture as the benchmark in this section. The overall architecture of SRCNN is shown in Fig. 2. As established in many traditional methods, for simplicity, SRCNN only implements the luminance components for training. SRCNN is a three-layer CNN, where the filter sizes of each layer are $64 \times 1 \times 9 \times 9$, $32 \times 64 \times 5 \times 5$

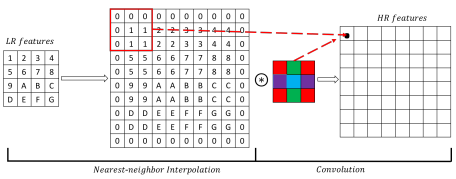


Figure 3: Sketch of the deconvolution layer used in FSRCNN [48], where ⊛ denotes the convolution operator.

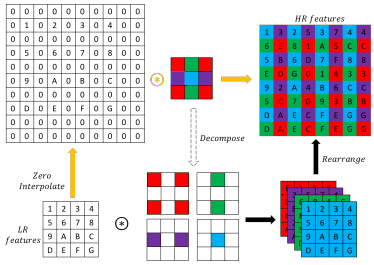


Figure 4: Detailed sketch of ESPCN [49]. The top process with the yellow arrow depicts the ESPCN from the view of zero interpolation, while the bottom process with the black arrow is the original ESPCN; ⊛ denotes the convolution operator.

and $1 \times 32 \times 5 \times 5$. The functions of these three nonlinear transformations are patch extraction, nonlinear mapping and reconstruction. The loss function for optimizing SRCNN is the mean square error (MSE), which will be discussed in the next section.

The formulation of SRCNN is relatively simple and can be envisioned as an ordinary CNN that approximates the complex mapping between the LR and HR spaces in an end-to-end manner. SRCNN reportedly demonstrated vast superiority over concurrent traditional methods, and we argue that its acclaim is owing to the CNN's strong capability of learning valid representations from big data in an end-to-end manner.

Despite the success of SRCNN, the following problems have inspired more effective architectures:

1) The input of SRCNN is the bicubic LR, an approximation of HR. However, these interpolated inputs have three drawbacks: (a) detail-smoothing effects introduced by these inputs may lead to further wrong estimations of the image structure; (b) employing interpolated versions as input is very time-consuming; and (c) when the downsampling kernel is unknown, one specific interpolated input as a raw estimation is unreasonable. Therefore, the first question emerges: can we design CNN architectures that directly implement LR as input to address these problems?[1]

2) The SRCNN is just a three-layer architecture. Can more complex CNN architectures (with different depths, widths and topologies) achieve better results? If yes, then how can we design such models of greater complexity?

3) The prior terms in the loss function that reflect properties of HR images are trivial. Can we integrate any property of the SISR process into the design of the CNN frame or other parts in the algorithms for SISR? If yes, then can these deep architectures with SISR properties be more effective in addressing some challenging SISR problems, such as the large scale factor SISR and the unknown downsampling of SISR?

Based on some solutions to these three questions, recent studies on deep architectures for SISR will be discussed in Sections III-B1, III-B2 and III-B3.

B. State-of-the-Art Deep SISR Networks

1) Learning Effective Upsampling with CNN: One solution to the first question is to design a module in the CNN architecture that adaptively increases the resolution. Convolution with pooling and stride convolution are the common downsampling operators in the basic CNN architecture. Naturally, people can implement the upsampling operation, which is known as deconvolution [50] or transposed convolution [51]. Given the upsampling factor, the deconvolution layer is composed of an arbitrary interpolation operator (usually, we choose the nearest neighbor interpolation for simplicity) and a following convolution operator with a stride of 1, as shown in Fig. 3. Readers should be aware that such deconvolution may not completely recover the information missing from convolution with pooling or stride convolution. Such a deconvolution layer has been successfully adopted in the context of network visualization [52], semantic segmentation [53] and generative modeling [54]. For a more detailed illustration of the deconvolution layer, readers can refer to [55]. To the best of our knowledge, FSRCNN [56] is the first work using this normal deconvolution layer to reconstruct HR images from LR feature maps. As mentioned previously, the usage of the deconvolution layer has two main advantages: one is that a reduction in computation is achieved because we just need to increase resolution at the end of the network; the other is that when the downsampling kernel is unknown, many reports, e.g., [57], have shown that when an inaccurate estimation is input, there are side effects on the final performance.

Although the normal deconvolution layer, which has already been involved in popular open source packages such as Caffe [58] and TensorFlow [59], offers a reasonably good solution to the first question, there is still an underlying problem: when we use the nearest neighbor interpolation, the points in the upsampled features are repeated several times in each direction. This configuration of the upsampled pixels is redundant. To circumvent this problem, Shi *et al.* proposed an efficient subpixel convolution layer in [49], known as ESPCN; the structure of ESPCN is shown in Fig. 4. Rather than increasing resolution by explicitly enlarging feature maps as the deconvolution layer does, ESPCN expands the channels of the output features for storing the extra points to increase resolution and then rearranges these points to obtain the HR output through a specific mapping criterion. As the expansion is carried out in the channel dimension, a smaller kernel size is sufficient. [55] further shows that when the ordinary but

[1]Generally, the first problem can be grouped into the third problem below. Because the solutions to this problem form the basis of many other models, it is necessary to introduce this problem separately as the first drawback.

redundant nearest neighbor interpolation is replaced with the interpolation that pads the subpixels with zeroes, the deconvolution layer can be simplified into the subpixel convolution in ESPCN. Obviously, compared with the nearest neighbor interpolation, this interpolation is more efficient, which can also verify the effectiveness of ESPCN.

2) The Deeper, The Better: In the DL research, there is theoretical work [60] showing that the solution space of a DNN can be expanded by increasing its depth or its width. In some situations, to attain more hierarchical representations more effectively, many works mainly focus on improvements acquired by increasing the depth. Recently, various DL-based applications have also demonstrated the great power of very deep neural networks despite many training difficulties. VDSR [61] is the first very deep model used in SISR. As shown in Fig. 5(a), VDSR is a 20-layer VGG-net [62]. The VGG architecture sets all kernel sizes as 3×3 (the kernel size is usually odd and takes the increase in the receptive field into account, and 3×3 is the smallest kernel size). To train this deep model, the authors used a relatively high initial learning rate to accelerate convergence and used gradient clipping to prevent the annoying gradient explosion problem.

In addition to the innovative architecture, VDSR has made two more contributions. The first one is that a single model is used for multiple scales since the SISR processes with different scale factors have a strong relationship with each other. This fact is the basis of many traditional SISR methods. Similar to SRCNN, VDSR takes the bicubic of LR as input. During training, VDSR puts the bicubics of LR of different scale factors together for training. For larger scale factors ($\times 3, \times 4$), the mapping for a smaller scale factor ($\times 2$) may also be informative. The second contribution is the residual learning. Unlike the direct mapping from the bicubic version to HR, VDSR uses deep CNN to learn the mapping from the bicubic to the residual between the bicubic and HR. The authors argued that residual learning could improve performance and accelerate convergence.

The convolution kernels in the nonlinear mapping part of VDSR are very similar, and in order to reduce parameters, Kim *et al.* further proposed DRCN [63], which utilizes the same convolution kernel in the nonlinear mapping part 16 times, as shown in Fig. 5(b). To overcome the difficulties of training a deep recursive CNN, a multisupervised strategy is applied, and the final result can be regarded as the fusion of 16 intermediate results. The coefficients for fusion are a list of trainable positive scalars with the summation of 1. As they showed, DRCN and VDSR have a quite similar performance.

Here, we believe that it is necessary to emphasize the importance of the multisupervised training in DRCN. This strategy not only creates short paths through which the gradients can flow more smoothly during backpropagation but also guides all the intermediate representations to reconstruct raw HR outputs. Finally, fusing all these raw HR outputs produces a wonderful result. However, for fusion, this strategy has two flaws: 1) once the weight scalars are determined in the training process, they will not change with different inputs; and 2) using a single scalar to weight HR outputs does not take pixelwise differences into consideration, that is, it would be better to weight different parts distinguishingly in an adaptive way.

It is hard to go deeper with a plain architecture such as VGG-net. Various deep models based on skip-connections can be extremely deep and have achieved state-of-the-art performance in many tasks. Among them, ResNet [64], [65], proposed by He *et al.*, is the most representative model. Readers can refer to [66], [67] for further discussions on why ResNet works well. In [68], the authors proposed SRResNet, which is composed of 16 residual units (a residual unit consists of two nonlinear convolutions with residual learning). In each unit, batch normalization (BN) [69] is used to stabilize the training process. The overall architecture of SRResNet is shown in Fig. 5(c). Based on the original residual unit in [65], Tai *et al.* proposed DRRN [70], in which basic residual units are rearranged in a recursive topology to form a recursive block, as shown in Fig. 5(d). Then, to accommodate parameter reduction, each block shares the same parameters and is reused recursively, such as in the single recursive convolution kernel in DRCN.

EDSR [71] was proposed by Lee *et al.* and has currently achieved state-of-the-art performance. EDSR has mainly made three improvements on the overall frame: 1) Compared with the residual unit used in previous work, EDSR removes the usage of BN, as shown in Fig. 5(e). The original ResNet with BN was designed for classification, where inner representations are highly abstract, and these representations can be insensitive to the shift introduced by BN. Regarding image-to-image tasks such as SISR, since the input and output are strongly related, if the convergence of the network is not a problem, then such a shift may harm the final performance. 2) Except for regular depth increasing, EDSR also increases the number of output features of each layer on a large scale. To relinquish the difficulties of training such a wide ResNet, the residual scaling trick proposed in [72] is employed. 3) Additionally, inspired by the fact that the SISR processes with different scale factors have strong relationships with each other, when training the models for $\times 3$ and $\times 4$ scales, the authors of [71] initialized the parameters with the pretrained $\times 2$ network. This pretraining strategy accelerates the training and improves the final performance.

The effectiveness of the pretraining strategy in EDSR implies that models for different scales may share many intermediate representations. To explore this idea further, similar to building a multiscale architecture as VDSR does on the condition of bicubic input, the authors of EDSR proposed MDSR to achieve the multiscale architecture, as shown in Fig. 5(g). In MDSR, the convolution kernels for nonlinear mapping are shared across different scales, where only the front convolution kernels for extracting features and the final subpixel upsampling convolution are different. At each update during training MDSR, minibatches for $\times 2$, $\times 3$ and $\times 4$ are randomly chosen, and only the corresponding parts of MDSR are updated.

In addition to ResNet, DenseNet [73] is another effective architecture based on skip connections. In DenseNet, each layer is connected with all the preceding representations, and the bottleneck layers are used in units and blocks to reduce the parameter amounts. In [74], the authors pointed

out that ResNet enables feature re-usage while DenseNet enables new feature exploration. Based on the basic DenseNet, SRDenseNet [75], as shown in Fig. 5(f), further concatenates all the features from different blocks before the deconvolution layer, which is shown to be effective in improving performance. MemNet [76], proposed by Tai *et al.*, uses the residual unit recursively to replace the normal convolution in the block of the basic DenseNet and adds dense connections among different blocks, as shown in Fig. 5(h). The authors explained that the local connections in the same block resemble the short-term memory and the connections with previous blocks resemble the long-term memory [77]. Recently, RDN [78] was proposed by Zhang *et al.* and uses a similar structure. In an RDN block, basic convolution units are densely connected similar to DenseNet, and at the end of an RDN block, a bottleneck layer is used, following with the residual learning across the whole block. Before entering the reconstruction part, features from all previous blocks are fused by the dense connection and residual learning.

3) Combining Properties of the SISR Process with the Design of the CNN Frame: In this subsection, we discuss some deep frames whose architectures or procedures are inspired by some representative methods for SISR. Compared with the abovementioned NN-oriented methods, these methods can be better interpreted, and they sometimes are more sophisticated in addressing certain challenging cases for SISR.

Combining sparse coding with deep NN: The sparse prior in nature images and the relationships between the HR and LR spaces rooted from this prior were widely used for their great performance and theoretical support. SCN [79] was proposed by Wang *et al.* and uses the learned iterative shrinkage and thresholding algorithm (LISTA) [80], which produces an approximate estimation of sparse coding based on NN, to solve the time-consuming inference in traditional sparse coding SISR. They further introduced a cascaded version (CSCN) [81] that employs multiple SCNs. Previous works such as SRCNN tried to explain general CNN architectures with the sparse coding theory, which from today's view may be somewhat unconvincing. SCN combines these two important concepts innovatively and gains both quantitative and qualitative improvements.

Learning to ensemble by NN: Different models specialize in different image patterns of SISR. From the perspective of ensemble learning, a better result can be acquired by adaptively fusing various models with different purposes at the pixel level. Motivated by this idea, MSCN was proposed by Liu *et al.* [82] by developing an extra module in the form of a CNN, taking the LR as input and outputting several tensors with the same shape as the HR. These tensors can be viewed as adaptive elementwise weights for each raw HR output. By selecting NNs as the raw SR inference modules, the raw estimating parts and the fusing part can be optimized jointly. However, in MSCN, the summation of coefficients at each pixel is not 1, which may be slightly incongruous.

Deep architectures with progressive methodology: Increasing SISR performance progressively has been extensively studied previously, and many recent DL-based approaches also exploit it from various perspectives. Here, we mainly discuss three novel works within this scope: DEGREE [83], combining the progressive property of ResNet with traditional subband reconstruction; LapSRN [84], generating SR of different scales progressively; and PixelSR [85], leveraging conditional autoregressive models to generate SR pixel-by-pixel.

Compared with other deep architectures, ResNet is intriguing for its progressive properties. Taking SRResNet for example, one can observe that directly sending the representations produced by intermediate residual blocks to the final reconstruction part will also yield a quite good raw HR estimator. The deeper these representations are, the better the results that can be obtained. A similar phenomenon of ResNet applied in recognition is reported in [66]. DEGREE, proposed by Yang *et al.*, combines this progressive property of ResNet with the subband reconstruction of traditional SR methods [86]. The residues learned in each residual block can be used to reconstruct high-frequency details, resembling the signals from a certain high-frequency band. To simulate subband reconstruction, a recursive residual block is used. Compared with the traditional supervised subband recovery methods that need to obtain subband ground truth by diverse filters, this simulation with recursive ResNet avoids explicitly estimating intermediate subband components, benefiting from the end-to-end representation learning.

As mentioned above, models for small scale factors can be used for a raw estimator of a large scale SISR. In the SISR community, SISR under large scale factors (*e.g.,*$\times 8$) has been a challenging problem for a long time. In such situations, plausible priors are imposed to restrict the solution space. A straightforward way to address this is to gradually increase resolution by adding extra supervision on the auxiliary SISR process of the small scale. Based on this heuristic prior, LapSRN, proposed by Lai *et al.*, uses the Laplacian pyramid structure to reconstruct HR outputs. LapSRN has two branches: the feature extraction branch and the image reconstruction branch, as shown in Fig. 6. At each scale, the image reconstruction branch estimates a raw HR output of the present stage, and the feature extraction branch outputs a residue between the raw estimator and the corresponding ground truth as well as extracts useful representations for the next stage.

When faced with large scale factors with a severe loss of necessary details, some researchers suggest that synthesizing rational details can achieve better results. In this situation, deep generative models, which will be discussed in the next sections, could be good choices. Compared with the traditional independent point estimation of the lost information, conditional autoregressive generative models using conditional maximum likelihood estimation in directional graphical models gradually generate high-resolution images based on the previously generated pixels. PixelRNN [87] and PixelCNN [88] are recent representative autoregressive generative models. The current pixel in PixelRNN and PixelCNN is explicitly dependent on the left and top pixels that have already been generated. To implement such operations, novel network architectures are elaborated. PixelSR was proposed by Dahl *et al.* and first applies conditional PixelCNN to SISR. The overall architecture is shown in Fig. 7. The conditioning CNN takes LR

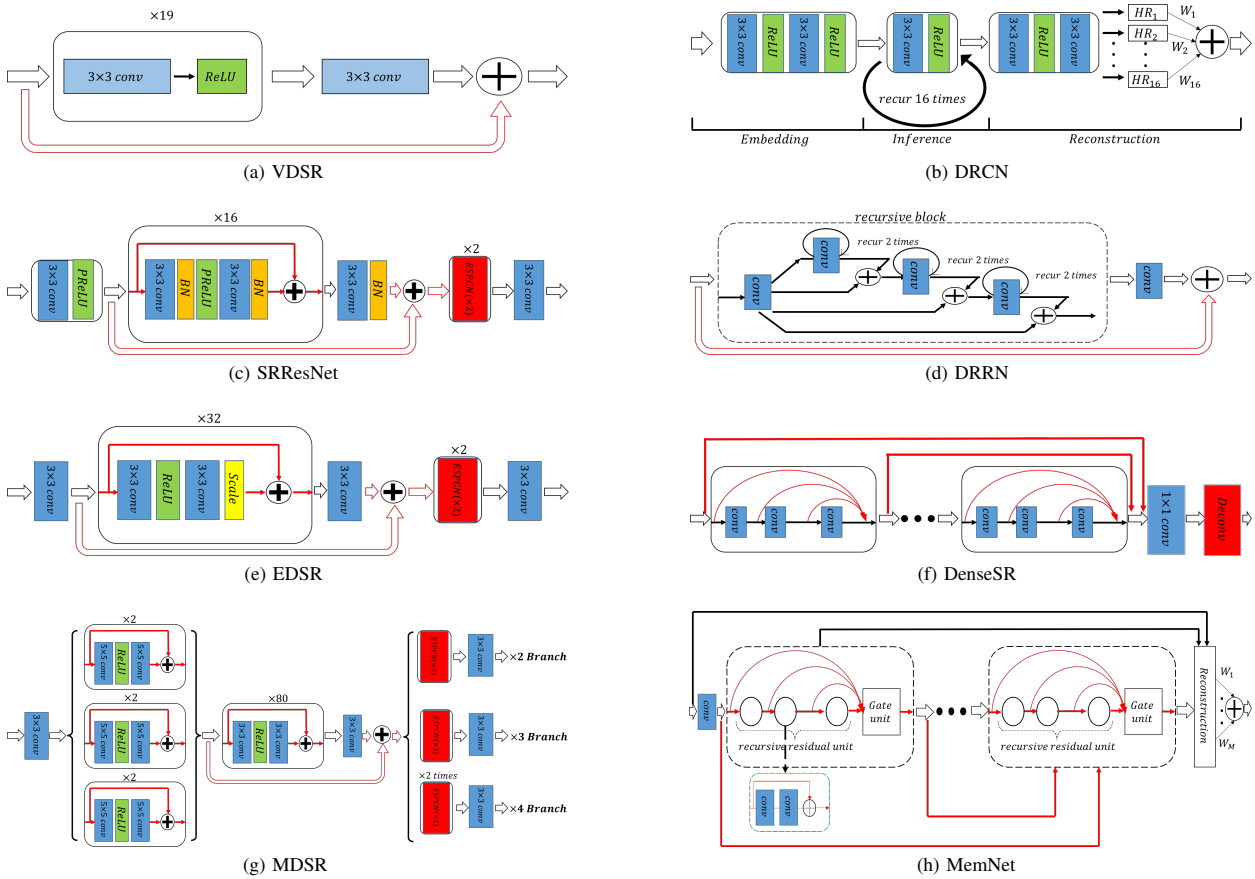


Figure 5: Sketch of several deep architectures for SISR.

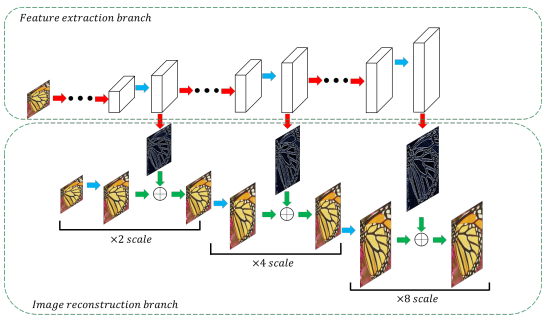


Figure 6: LapSRN architecture. Red arrows indicate the convolutional layer; blue arrows indicate transposed convolutions (upsampling); green arrows denote elementwise addition operators.

as input, which provides LR-conditional information to the whole model, and the PixelCNN part is the autoregressive inference part. The current pixel is determined by these two parts together using the current softmax probability:

$$P(y_i|x, y_{<i}) = \text{softmax}(A_i(x) + B_i(y_{<i})), \qquad (2)$$

where x is the LR input, y_i is the current HR pixel to be generated, $y_{<i}$ are the generated pixels, $A_i(\cdot)$ denotes the conditioning network predicting a vector of logit values corresponding to the possible values, and $B_i(\cdot)$ denotes the prior network predicting a vector of logit values of the ith output pixel. Pixels with the highest probability are taken as the final output pixel.

Similarly, the whole network is optimized by minimizing cross-entropy loss (maximizing the corresponding log-likelihood) between the model's prediction and the discrete ground-truth labels.

Deep architectures with backprojection: Iterative back-projection [89] is an early SR algorithm that iteratively computes the reconstruction error and then feeds it back to tune the HR results. Recently, DBPN [90], proposed by Haris *et al.*, uses deep architectures to simulate iterative backprojection and further improves performance with dense connections [73], which is shown to achieve wonderful performance in the $\times 8$ scale. As shown in Fig. 8, the dense connection and 1×1 convolution for reducing the dimension is first applied across different up-projection (down-projection) units; next, in the tth up-projection unit, the current LR feature input $L^{\widetilde{t-1}}$ is first deconvoluted to obtain a raw HR feature H_0^t, and H_0^t is

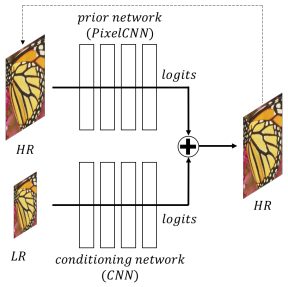


Figure 7: Sketch of the pixel recursive SR architecture.

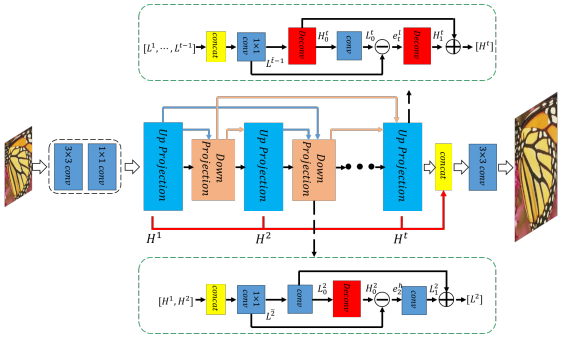


Figure 8: Sketch of the DBPN architecture.

backprojected to the LR feature L_0^t. The residue between two LR features $e_t^l = L^{t-1} - L_0^{t-1}$ is then deconvoluted and added to H_0^t to obtain a finer HR feature H^t. The down-projection unit is defined very similarly in an inverse way.

Usage of additional information from LR: Although modern deep NNs are skillful in extracting various ranges of useful representations in end-to-end manners, in some cases, it is still helpful to select some information to process explicitly. For example, the DEGREE [83] takes the edge map of LR as another input. Recent studies tend to use more complex information of LR directly, two examples of which are the following: SFT-GAN [91], with extra semantic information of LR for better perceptual quality, and SRMD [92], incorporating degradation into input for multiple degradations.

[93] reported that using a semantic prior helps improve the performance of many SISR algorithms. Leveraging powerful deep architectures recently designed for segmentation, Wang *et al.* [91] used semantic segmentation maps of interpreted LR as additional input and deliberated the spatial feature transformation (SFT) layer to handle them. With this extra information from high-level tasks, the proposed work is more skilled in generating textual details.

To take degradations of different LRs into account, SRMD first applied a parametric zero-mean anisotropic Gaussian kernel to stand for the blur kernel and the additive white Gaussian noise with hyperparameter ρ^2 to represent noise. Then, a simple regression is used to obtain its covariance matrix. These sufficient statistics are dimensionally stretched to concatenate with LR in the channel dimension, and with such input, a deep model is trained. Notably, when SRMD is tested with real images, the needed parameters on the degradation level are obtained by grid search.

Reconstruction-based frameworks based on priors offered by deep NN: Sophisticated priors are of key points for efficient reconstruction-based SISR algorithms to address different cases flexibly. Recent works showed that deep NNs could provide well-performing priors mainly from two perspectives: priors in the deep NN learn from data in advance within a plug-and-play approach and direct reconstruction of output, leveraging intriguing but still unclear priors of deep architectures themselves.

Given the degraded version y, the reconstruction-based algorithms aim to obtain the desired result $\hat{x}$ by solving

$$\hat{x} = \arg\min ||Hx - y||_2^2 + R(x), \tag{3}$$

where H is the degradation matrix and $R(x)$ is regularization, also called a prior from the Bayesian view. [94] split (3) into a data part and a prior part with variable splitting techniques and then replaced the prior part with efficient denoising algorithms. Regarding different degradation cases, one only needs to change denoising algorithms for the prior part, behaving in so-called plug-and-play manners. Recent works [95], [96], [97] use deep discriminatively trained NNs under different noise levels as denoisers in various inverse problems, and IRCNN [96] is the first one among them to address SISR. In IRCNN, they first trained a series of CNN-based denoisers with different noise levels, and took backprojection as the reconstruction part. The LR is first preceded by several backprojection iterations and then denoised by CNN denoisers with decreasing noise levels along with backprojection. The iteration number is empirically set to 30. In IRCNN, the authors use deep networks to learn a set of image priors and then plug the priors into the reconstruction framework; the experimental results in these cases are better than the contemporary methods that only employ example-based training.

Recently, Ulyanov *et al.* showed in [98] that the structure of deep neural networks could capture a considerable amount of low-level image statistical priors. They reported that when neural networks are used to fit images of different statistical properties, the convergence speed for different kinds of images can also be different. As shown in Fig. 9, natural-looking images, whose different parts are highly relevant, will converge much faster. In contrast, images such as noises and shuffled images, which have little inner relationship, tend to converge more slowly. Many inverse problems such as denoising and super-resolution are modeled as the pixel-wise summation of the original image and the independent additive noises. Based on the observed prior, when used to fit these degraded images, the neural networks tend to fit the natural-looking images first, which can be used to retain the natural-looking parts as well as to filter the noisy ones. To illustrate the effectiveness of the proposed prior for SISR, only given the LR x_0, the authors took a fixed random vector z as input to fit the HR x with a randomly initialized DNN f_θ by optimizing

$$\min_\theta ||d(f_\theta(z)) - x_0||_2^2, \tag{4}$$

where $d(\cdot)$ is a common differentiable downsampling operator. The optimization is terminated in advance for only filtering

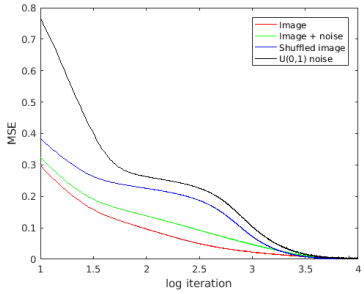


Figure 9: Learning curves for the reconstruction of different kinds of images. We re-implement the experiment in [98] with the image 'butterfly' in Set5.

noisy parts. Although these totally unsupervised methods are outperformed by other supervised learning methods, they perform considerably better than some other naive methods.

Deep architectures with internal examples: Internal-example SISR algorithms are based on the recurrence of small pieces of information across different scales of a single image, which are shown to be better at addressing specific details rarely existing in other external images [99]. ZSSR [100], proposed by Shocher *et al.*, is the first literature combining deep architectures with internal-example learning. In ZSSR, other than the image for testing, no extra images are needed, and all the patches for training are taken from different degraded pairs of the test image. As demonstrated in [101], the visual entropy inside a single image is much smaller than the large training dataset collected from wide ranges, so unlike external-example SISR algorithms, a very small CNN is sufficient. As we mentioned previously for VDSR, the training data for a small-scale model can also be useful for training large-scale models. Additionally, based on this trick, ZSSR can be more robust by collecting more internal training pairs with small scale factors for training large-scale models. However, this approach will increase runtime immensely. Notably, when combined with the kernel estimation algorithms mentioned in [102], ZSSR performs quite well with the unknown degradation kernels.

Recently, Tirer *et al.* argued that degradation in LR decreases the performance of internal-example algorithms [103]. Therefore, they proposed to use reconstruction-based deep frame IDBP [97] to obtain an initial SR result and then conduct internal-example-based network training similar to ZSSR. This method was believed to combine two successful techniques that address the mismatch between training and test, and it has achieved robust performance in these cases.

C. Comparisons among Different Models and Discussion

In this section, we will summarize recent progress in deep architectures for SISR from two perspectives: quantitative comparisons for those trained by specific blurring, and comparisons on those models for handling nonspecific blurring.

For the first part, quantitative criteria mainly include the following:

1) PSNR/SSIM [104] for measuring reconstruction quality: Given two images I and $\hat{I}$ both with N pixels, the MSE and peak signal-to-noise ratio (PSNR) are defined as

$$MSE = \frac{1}{N}||I - \hat{I}||_F^2, \tag{5}$$

$$PNSR = 10\log_{10}^{(\frac{L^2}{MSE})}, \tag{6}$$

where $||\cdot||_F^2$ is the Frobenius norm and L is usually 255. The structural similarity index (SSIM) is defined as

$$SSIM(I,\hat{I}) = \frac{2\mu_I\mu_{\hat{I}} + k_1}{\mu_I^2 + \mu_{\hat{I}}^2 + k_1} \cdot \frac{\sigma_{I\hat{I}} + k_2}{\sigma_I^2 + \sigma_{\hat{I}}^2 + k_2}, \tag{7}$$

where μ_I and σ_I^2 is the mean and variance of I, $\sigma_{I\hat{I}}$ is the covariance between I and $\hat{I}$, and k_1 and k_2 are constant relaxation terms.

2) Number of parameters of NN for measuring storage efficiency (Params).

3) Number of composite multiply-accumulate operations for measuring computational efficiency (Mult&Adds): Since operations in NNs for SISR are mainly multiplications with additions, we use Mult&Adds in CARN [105] to measure computation, assuming that the desired SR is 720p.

Notably, it has been shown in [48] and [49] that the training datasets have a great influence on the final performance, and usually, more abundant training data will lead to better results. Generally, these models are trained via three main datasets: 1) 91 images from [19] and 200 images from [106], called the 291 dataset (some models only use 91 images); 2) images derived from ImageNet [107] randomly; and 3) the newly published DIV2K dataset [108]. In addition to the different number of images each dataset contains, the quality of images in each dataset is also different. Images in the 291 dataset are usually small (on average, 150×150), images in ImageNet are much larger, while images in DIV2K are of very high quality. Because of the restricted resolution of the images in the 291 dataset, models on this set have difficulties in obtaining large patches with large receptive fields. Therefore, models based on the 291 dataset usually take the bicubic of LR as input, which is quite time-consuming. Table I compares different models on the mentioned criteria.

From Table I, we can see that generally as the depth and the number of parameters grow, the performance improves. However, the growth rate of performance levels off. Recently, some works on designing light models [109], [105], [110] and learning sparse structural NN [111] were proposed to achieve relatively good performance with less storage and computation, which are very meaningful in practice.

For the second part, we mainly show that the performance of the models for some specific degradation dropped drastically when the true degradation mismatches the one assumed for training. For example, we use four models, including EDSR trained with bicubic degradation [71], IRCNN [96], SRMD [92] and ZSSR [100], to address LRs generated by Gaussian kernel degradation (kernel size of 7×7 with bandwidth 1.6), as shown in Fig. 10, and the performance of EDSR dropped drastically with obvious blur, while other models for nonspecific degradation perform quite well. Therefore, to

Table I: Comparisons among some representative deep models.

Models	PSNR/SSIM(×4)	Train data	Parameters	Mult&Adds
SRCNN_EX [48]	30.49/0.8628	ImageNet subset	57K	52.5G
ESPCN [49]	30.90/-	ImageNet subset	20K	1.43G
VDSR [61]	31.35/0.8838	G200+Yang91	665K	612.6G
DRCN [63]	31.53/0.8838	Yang91	1.77M(recursive)	17974.3G
DRRN [70]	31.68/0.8888	G200+Yang91	297K(recursive)	6796.9G
LapSRN [84]	31.54/0.8855	G200+Yang91	812K	29.9G
SRResNet [68]	32.05/0.9019	ImageNet subset	1.5M	127.8G
MemNet [76]	31.74/0.8893	G200+Yang91	677K(recursive)	2265.0G
RDN [78]	32.61/0.9003	DIV2K	22.6M	1300.7G
EDSR [71]	32.62/0.8984	DIV2K	43M	2890.0G
MDSR [71]	32.60/0.8982	DIV2K	8M	407.5G
DBPN [90]	32.47/0.898	DIV2K+Flickr+ ImageNet subset	10M	5715.4G

address some longstanding problems in SISR, such as unknown degradation, the direct usage of general deep learning techniques may not be sufficient. More effective solutions can be achieved by combining the power of DL and the specific properties of the SISR scene.

IV. Optimization Objectives for DL-based SISR

A. Benchmark of Optimization Objectives for DL-based SISR

We select the MSE loss used in SRCNN as the benchmark. It is known that using MSE favors a high PSNR, and PSNR is a widely used metric for quantitatively evaluating image restoration quality. Optimizing MSE can be viewed as a regression problem, leading to a point estimation of θ as

$$\min_{\theta} \sum_{i} ||F(x_i;\theta) - y_i||^2, \tag{8}$$

where (x_i, y_i) are the ith training examples and $F(x;\theta)$ is a CNN parameterized by θ. Here, (8) can be interpreted in a probabilistic way by assuming Gaussian white noise ($\mathcal{N}(\epsilon; 0, \sigma^2 I)$) independent of the image in the regression model, and then, the conditional probability of y given x becomes a Gaussian distribution with mean $F(x;\theta)$ and the diagonal covariance matrix $\sigma^2 I$, where I is the identity matrix:

$$p(y|x) = \mathcal{N}(y; F(x;\theta), \sigma^2 I). \tag{9}$$

Then, using maximum likelihood estimation (MLE) on the training examples with (9) will lead to (8).

The Kullback-Leibler divergence (KLD) between the conditional empirical distribution P_{data} and the conditional model distribution P_{model} is defined as

$$D_{KL}(P_{data}||P_{model}) = E_{z\sim P_{data}}[\log \frac{P_{data}(z)}{P_{model}(z)}]. \tag{10}$$

We call (10) the forward KLD, where $z = y|x$ denotes the HR (SR) conditioned on its LR counterpart, P_{data} and P_{model} are the conditional distributions of $HR|LR$ and $SR|LR$, respectively, where $E_{x\sim P_{data}}[\log P_{data}(z)]$ is an intrinsic term determined by the training data regardless of the parameter θ of the model (or the model distribution $P_{model}(x;\theta)$). Hence, when we use the training samples to estimate parameter θ, minimizing this KLD is equivalent to MLE.

Here, we have demonstrated that MSE is a special case of MLE, and MLE is a special case of KLD. However, we may conjecture whether the assumptions underlying these specializations are violated. This consideration has led to some emerging objective functions from four perspectives:

1) Translating MLE into MSE can be achieved by assuming Gaussian white noise. Although the Gaussian model is the most widely used model for its simplicity and technical support, what if this independent Gaussian noise assumption is violated in a complicated scene such as SISR?

2) To use MLE, we need to assume the parametric form of the data distribution. What if the parametric form is misspecified?

3) Apart from KLD in (10), are there any other distances between probability measures that we can use as the optimization objectives for SISR?

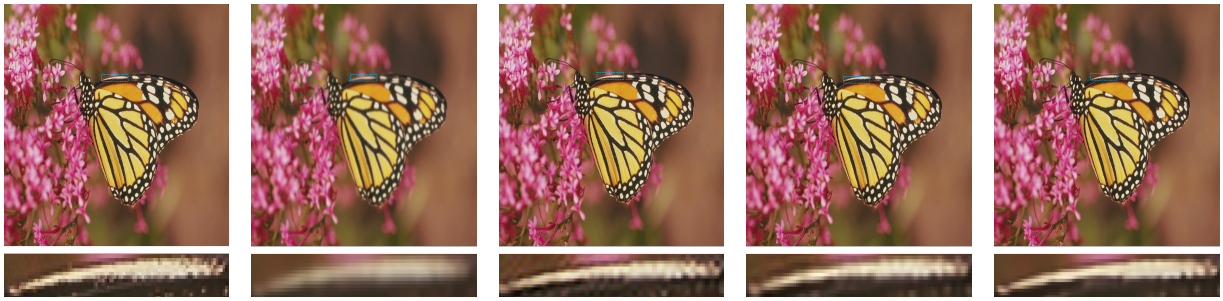
4) Under specific circumstances, how can we choose the suitable objective functions according to their properties?

Based on some solutions to these four questions, recent work on optimization objectives for DL-based SISR will be discussed in Sections IV-B, IV-C, IV-D and IV-E, respectively.

B. Objective Functions Based on non-Gaussian Additive Noises

The poor perceptual quality of the SISR images obtained by optimizing MSE directly demonstrates a fact: using Gaussian additive noise in the HR space is not good enough. To address this problem, solutions are proposed from two aspects: use other distributions for this additive noise, or transfer the HR space to some space where the Gaussian noise is reasonable.

1) Denote Additive Noise with Other Probability Distributions: In [112], Zhao *et al.* investigated the difference between



(a) HR (b) EDSR(27.80dB/0.9012) (c) IRCNN(34.63dB/0.9548) (d) ZSSR(30.45dB/0.9384) (e) SRDM(37.71dB/0.9723)

Figure 10: Comparisons of 'monarch' in Set14 for scale 2 with Gaussian kernel degradation. We can see that, given the degradation mismatch with that of training, the performance of EDSR decreases drastically.

mean absolute error (MAE) and MSE used to it optimize NN in image processing. Similar to (8), MAE can be written as

$$\min_{\theta} \sum_{i} ||F(x_i;\theta) - y_i||_1. \tag{11}$$

From the perspective of probability, (11) can be interpreted as introducing Laplacian white noise, and similar to (9), the conditional probability becomes

$$p(y|x) = Laplace(y; F(x;\theta), bI). \tag{12}$$

Compared with MSE in regression, MAE is believed to be more robust against outliers. As reported in [112], when MAE is used to optimize an NN, the NN tends to converge faster and produce better results. The authors argued that the reason might be because MAE could guide NN to reach a better local minimum. Other similar loss functions in robust statistics can be viewed as modeling additive noises with other probability distributions.

Although these specific distributions often cannot represent unknown additive noise very precisely, their corresponding robust statistical loss functions are used in many DL-based SISR works for their conciseness and advantages over MSE.

2) Using MSE in a Transformed Space: Alternatively, we can search for a mapping Ψ to transform the HR space to some space where Gaussian white noise can be used reasonably. From this perspective, Bruna *et al.* [113] proposed so-called perceptual loss to leverage deep architectures. In [113], the conditional probability of the residual r between HR and LR given the LR x is stimulated by the Gibbs energy model:

$$p(r|x) = \exp(-||\Phi(x) - \Psi(r)||^2 - \log Z), \tag{13}$$

where Φ and Ψ are two mappings between the original spaces and the transformed ones, and Z is the partition function. The features produced by sophisticated supervised deep architectures have been shown to be perceptually stable and discriminative, denoted by $\Psi(r)$[2]. Then, Ψ represents the corresponding deep architectures. In contrast, Φ is the mapping between the LR space and the manifold represented by $\Psi(r)$, trained by minimizing the Euclidean distance as

$$\min_{\Phi} ||\Phi(x) - \Psi(r)||^2. \tag{14}$$

After Φ is obtained, the final result r can be inferred with SGD by solving

$$\min_{r} ||\Phi(x) - \Psi(r)||^2. \tag{15}$$

For further improvement, [113] also proposed a fine-tuning algorithm in which Φ and Ψ can be fine-tuned to the data. Similar to the alternating updating in GAN, Φ and Ψ are fine-tuned with SGD based on the current r. However, this fine-tuning will involve calculating the gradient of the partition function Z, which is a well-known difficult decomposition into the positive phase and the negative phase of learning. Hence to avoid sampling within inner loops, a biased estimator of this gradient is chosen for simplicity.

The inference algorithm in [113] is extremely time-consuming. To improve efficiency, Johnson *et al.* utilized this perceptual loss in an end-to-end training manner [114]. In [114], the SISR network is directly optimized with SGD by minimizing the MSE in the feature manifold produced by VGG-16 as follows:

$$\min_{\theta} ||\Psi(F(x;\theta)) - \Psi(y)||^2, \tag{16}$$

where Ψ is the mapping represented by VGG-16, $F(x;\theta)$ denotes the SISR network, and y is the ground truth. Compared with [113], [114] replaces the nonlinear mapping Φ and the expensive inference with an end-to-end trained CNN, and their results show that this change does not affect the restoration quality but does accelerate the whole process.

Perceptual loss mitigates blurring and leads to more visually-pleasing results compared with directly optimizing MSE in the HR space. However, there remains no theoretical analysis on why this approach works. In [113], the author generally concluded that successful supervised networks used for high-level tasks could produce very compact and stable features. In these feature spaces, small pixel-level variation and much other trivial information can be omitted, making these feature maps mainly focus on pixels of human interest. At

[2]Either the scattering network or VGG can be denoted by Ψ. When Ψ is VGG, there is no residual learning and fine-tuning.

the same time, with the deep architectures, the most specific and discriminative information of the input is shown to be retained in feature spaces because of the great performance of the models applied in various high-level tasks. From this perspective, using MSE in these feature spaces will focus more on the parts that are attractive to human observers with little loss of original contents, so perceptually pleasing results can be obtained.

C. Optimizing Forward KLD with Nonparametric Estimation

Parametric estimation methods such as MLE need to specify in advance the parametric form the distribution of data, which suffers from model misspecification. Different from parametric estimation, nonparametric estimation methods such as kernel distribution estimation (KDE) fit the data without distributional assumptions, which are robust when the real distributional form is unknown. Based on nonparametric estimation, recently, the contextual loss [115], [116] was proposed by Mechrez *et al.* to maintain natural image statistics. In the contextual loss, a Gaussian kernel function is applied:

$$K(x,y) = \exp(-dist(x,y)/h - \log Z), \tag{17}$$

where $dist(x,y)$ can be any symmetric distance between x and y, h is the bandwidth, and the partition function $Z = \int \exp(-dist(x,y)/h)dy$. Then, P_{data} and P_{model} are

$$\begin{aligned} P_{data}(z) &= \sum_{z_i \sim P_{data}} K(z, z_i), \\ P_{model}(z) &= \sum_{w_j \sim P_{model}} K(z, w_j), \end{aligned} \tag{18}$$

and (10) can be rewritten as

$$\begin{aligned} &D_{KL}(P_{data}||P_{model}) = \\ &\frac{1}{N}\sum_k [\log \sum_{z_i \sim P_{data}} K(z_k, z_i) - \log \sum_{w_j \sim P_{model}} K(z_k, w_j)]. \end{aligned} \tag{19}$$

The first log term in (19) is a constant with respect to the model parameters. Let us denote the kernel $K(z_k, w_j)$ in the second log term by A_{kj}. Then, the optimization objective in (19) can be rewritten as

$$-\frac{1}{N}\sum_k \log \sum_j A_{kj}. \tag{20}$$

With the Jensen inequality, we can obtain a lower bound of (20):

$$-\frac{1}{N}\sum_k \log \sum_j A_{kj} \geq -\log \frac{1}{N}\sum_k \sum_j A_{kj} \geq 0. \tag{21}$$

The first equality holds if and only if $\forall k, k'$, $\sum_j A_{kj} = \sum_j A_{k'j}$. Both equalities hold if and only if $\forall k$, $\sum_j A_{kj} = 0$. When (20) reaches 0, the given lower bound also reaches 0. Therefore, we can take this lower bound as the optimization objective alternatively.

We can further simplify the lower bound in (21). The lower bound can be rewritten as

$$-\log \frac{1}{N}\sum_j ||A_j||_1, \tag{22}$$

where $A_j = (A_{1j}, \cdots, A_{kj})^T$, and $\|\cdot\|_1$ is the ℓ_1 norm. When the bandwidth $h \to 0$, the affinity A_{kj} will degrade into the indicator function, which means if $x_k = y_j$, $A_{kj} \approx 1$; otherwise, $A_{kj} \approx 0$. In this case, the ℓ_1 norm can be approximated well by the ℓ_∞ norm, which returns the maximum element of the vector. Thus, (22) can degenerate into the contextual loss in [115], [116]:

$$-\log \frac{1}{N}\sum_j \max_k A_{kj}. \tag{23}$$

Recently, implicit likelihood estimation (IMLE) [117] was proposed and its conditional version was applied to SISR [118]. Here, we will briefly show that minimizing IMLE equals minimizing an upper bound of the forward KLD with KDE. Let us use a Gaussian kernel as

$$K(x,y) = \frac{1}{\sqrt{2\pi}h}\exp\left(-\frac{\|x-y\|_2^2}{2h^2}\right). \tag{24}$$

As with (20), the optimization objective can be rewritten as

$$-\frac{1}{N}\sum_k \log \sum_j e^{-\frac{\|z_k - w_j\|_2^2}{2h^2}}. \tag{25}$$

With $\{w_j\}_{j=1}^m$ and $\{z_k\}_{k=1}^N$, we can obtain a simple upper bound of (25) as

$$\begin{aligned} &-\frac{1}{N}\sum_k \log\left(m \min_j e^{-\frac{\|z_k - w_j\|_2^2}{2h^2}}\right) \\ &= \frac{1}{N}\sum_k (\min_j \frac{||z_k - w_j||_2^2}{2h^2} - \log m). \end{aligned} \tag{26}$$

Minimizing (26) equals minimizing

$$\sum_k \min_j \|z_k - w_j\|_2^2, \tag{27}$$

which is the core of the optimization objective of IMLE.

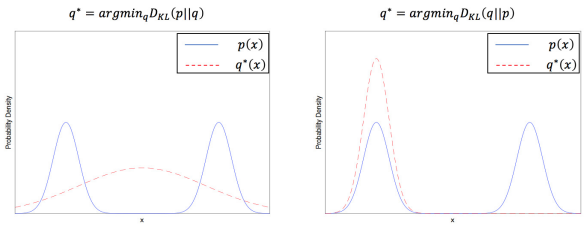
As above, the recently proposed contextual loss and IMLE are illustrated via nonparametric estimation and KLD. Visually pleasing results were reported using the contextual loss and IMLE. However, as KDE is generally very time-consuming, several reasonable approximations along with acceleration algorithms were applied.

D. Other Distances between Probability Measures Used in SISR

As KLD is an asymmetric (pseudo) distance for measuring similarity between two distributions, in this subsection, we begin with the inverse form of forward KLD, namely, backward KLD. The backward KLD is defined as

$$D_{KL}(P_{model}||P_{data}) = E_{z \sim P_{model}}[\log \frac{P_{model}(z)}{P_{data}(z)}]. \tag{28}$$

When $P_{model} = P_{data}$, both KLDs reach the minimum of 0. However, when the solution is inadequate, these two KLDs will lead to quite different results. Here, we use a toy example



(a) forward KLD (b) backward KLD

Figure 11: A toy example to illustrate the difference between the forward KLD and the backward KLD.

to illustrate a simple case of inadequate solutions, as shown in Fig. 11.

The unknown wanted distribution is a Gaussian mixture model (GMM) with two modes, denoted as $P(x)$, and we model it by a single Gaussian distribution. We can easily see that optimizing the forward KLD results in a solution locating at the middle areas of two modes, while optimizing the backward KLD makes the result close to the most prominent mode.

From Fig. 11 we can see that, under inadequate solutions, optimizing the forward KLD will lead to the well-known regression-to-the-mean problem, while optimizing the backward KLD only concentrates on the main modality. The former is one of the reasons for blurring, and some researchers [119] argued that the latter improves the visual quality but makes the results collapse to some patterns.

Different distances may lead to different results under an inadequate solution. Readers can refer to [120] for further understanding. In most low-level computer vision tasks, P_{data} is an empirical distribution and P_{model} is an intractable distribution. For this reason, the backward KLD is unpractical for optimizing deep architectures. To relieve optimizing difficulties, we replace the asymmetric KLD with the symmetric Jensen-Shannon divergence (JSD) as follows:

$$\begin{aligned} JS(P_{data}||P_{model}) = &\frac{1}{2}KL[P_{data}||\frac{P_{data}+P_{model}}{2}]+ \\ &\frac{1}{2}KL[P_{model}||\frac{P_{data}+P_{model}}{2}]. \end{aligned} \tag{29}$$

Optimizing (29) explicitly is also very difficult. Generative adversarial nets (GANs) proposed by Goodfellow *et al.* use the objective function below to implicitly address this problem in a game theory scenario, successfully avoiding the troubling approximate inference and approximation of the partition function gradient:

$$\min_G \max_D [E_{z\sim P_{data}} \log D(z) + E_{z\sim P_{model}} \log(1-D(z))], \tag{30}$$

where G is the main part called the generator supervised by an auxiliary part D called the discriminator. The two parts update alternatively, and when the discriminator cannot give useful information to the generator anymore, in other words, the outputs of the generator totally confuse the discriminator, the optimization procedure is completed. For the detailed discussion on GANs, readers can refer to [45]. Recent works have shown that sophisticated architectures and suitable hyperparameters can help GANs perform excellently. The representative works on GAN-based SISR are [68] and [121]. In [68], the generator of the GAN is the SRResNet mentioned previously, and the discriminator refers to the design criterion of DCGAN [54]. In the context of GANs, a recent work [121] follows a similar path except with a different architecture. Very recently, by leveraging the extension of the basic GAN framework [122], [123] was proposed as an unsupervised SR algorithm. Fig. 12 shows the results of the GAN and MSE with the same architecture; despite the lower PSNR due to artifacts, the visual quality improves by using the GAN for SISR.

Generally, GANs offer an implicit optimization strategy in an adversarial training way by using deep neural networks. Based on this, more rational but complicated measures such as Wasserstein distances [124], f-divergence [125][3] and maximum mean discrepancy (MMD) [126] are taken as alternatives to JSD for training GANs.

E. Characters of Different objective functions

Now, we can see that those losses mentioned in Section IV-B explicitly model the relation between LR and its HR counterpart. Here, we follow the methodology of [127] and call the losses that were based on measuring the dissimilarity between training pairs the distortion-aimed losses. When the training data are not sufficient, distortion losses usually ignore the particularity of data and appear ineffective to measure the similarity between the source and target distributions.

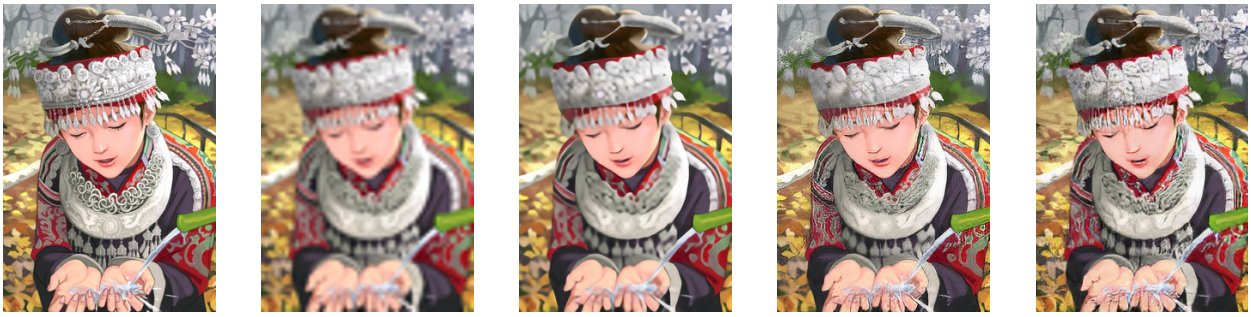
The losses mentioned in Sections IV-C and IV-D are rooted from measuring the similarity between distributions, which is thought to measure the perceptual quality. Here, we call them perception-aimed losses. Recently, Blau *et al.* [127] discussed the inherent trade-off between the two kinds of losses. Their discussion can be simplified into an optimization problem:

$$P(D) = \min_{P_{\hat{Y}|X}} d(P_Y, P_{\hat{Y}}) \ s.t. \ E[\Delta(Y,\hat{Y})] \leq D. \tag{31}$$

$\Delta(\cdot,\cdot)$ is distortion-aimed loss, and $d(\cdot,\cdot)$ is the (pseudo) distance between distributions. Furthermore, the author also proved that if $d(\cdot,\cdot)$ is convex in its second argument, then the $P(D)$ is monotonically nonincreasing and convex. From this property, we can draw the curve of $P(D)$ and easily see this trade-off, as shown in Fig. 13(a), such that improving one must be at the expense of the other. However, as shown in Section IV-B, using MSE in the VGG feature space achieves a better quality, and choosing suitable Δ and d may ease this trade-off.

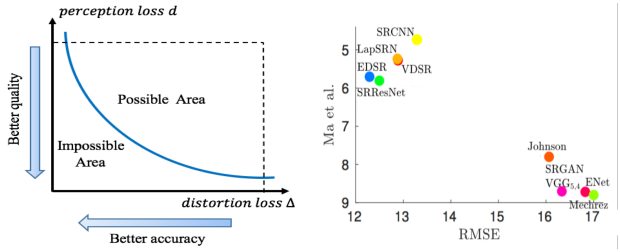
For the perception-aimed losses mentioned in Sections IV-C and IV-D, up to now, there has been no rigorous analysis on their differences. Here, we apply the nonreference quality assessment proposed by Ma *et al.* [95] with RMSE to conduct quantitative comparisons, and the representative qualitative comparisons are depicted in Fig. 13(b). To summarize, we

[3]Forward KLD, backward KLD and JSD can all be regarded as the special cases of f-divergence.



(a) HR (b) bicubic(21.59dB/0.6423) (c) SRResNet(23.53dB/0.7832) (d) SRGAN(21.15dB/0.6868) (e) SRCX(20.88dB/0.6002)

Figure 12: Visual comparisons between the MSE, MSE + GAN and MAE +GAN + Contextual Loss (The authors of [68] and [116] released their results.) We can see that the perceptual loss leads to a lower PSNR/SSIM but a better visual quality.



(a) Perception-distortion trade-off (b) Perception-distortion evaluation

Figure 13: (a) The perception-distortion space is divided by the perception-distortion curve, where an area cannot be attained. (b) Use of the nonreference metric proposed by [95] and RMSE to perform quantitative comparisons from the perception and distortion views; the included methods are [47], [84], [71], [61], [68], [121], [116], [114].

should be aware that there is no one-fits-all objective function, and we should choose one that is suitable to the context of an application.

V. Trends and Challenges

Along with the promising performance that DL algorithms have achieved in SISR, there remain several important challenges and inherent trends as follows.

1) Lighter Deep Architectures for Efficient SISR: Although the high accuracy of advanced deep models has been achieved for SISR, it is still difficult to deploy these models to real-world scenarios, which is mainly due to massive parameters and computation. To address this issue, we need to design light deep models or slim the existing deep models for SISR with fewer parameters and computation at the expense of little or no performance degradation. Hence, in the future, researchers are expected to focus more on reducing the size of NNs for speeding up the SISR process.

2) More Effective DL Algorithms for Large-scale SISR and SISR with Unknown Corruption: Generally, DL algorithms proposed in recent years have improved the performance of traditional SISR tasks by a large margin. However, the large scale of SISR and the SISR with unknown corruption, the two major challenges in the SR community, are still lacking very effective remedies. DL algorithms are thought to be skilled at addressing many inferences or unsupervised problems, which is of key importance to address these two challenges. Therefore, by leveraging the great power of DL, more effective solutions to these two demanding problems are expected.

3) Theoretical Understanding of Deep Models for SISR: The success of deep learning is said to be attributed to learning powerful representations. However, to date, we still cannot understand these representations very well, and the deep architectures are treated as a black box. For DL-based SISR, the deep architectures are often viewed as a universal approximation, and the learned representations are often omitted for simplicity. This behavior is not beneficial for further exploration. Therefore, we should not only focus on whether a deep model works but also concentrate on why and how it works. That is, more theoretical explorations are needed.

4) More Rational Assessment Criteria for SISR in Different Applications: In many applications, we need to design the desired objective function for a specific application. However, in most cases, we cannot give an explicit and precise definition to assess the requirement for the application, which leads to the vagueness of the optimization objectives. Many works, although for different purposes, simply employ MSE as the assessment criterion, which has been shown as a poor criterion in many cases. In the future, we think that it is of great necessity to make clear definitions for assessments in various applications. Based on these criteria, we can design better targeted optimization objectives and compare algorithms in the same context more rationally.

VI. Conclusion

This paper presents a brief review of recent deep learning algorithms on SISR. It divides the recent works into two categories: the deep architectures for simulating the SISR process and the optimization objectives for optimizing the whole process. Despite the promising results reported so far,

there are still many underlying problems. We summarize the main challenges into three aspects: the acceleration of deep models, the extensive comprehension of deep models and the criteria for designing and evaluating the objective functions. Along with these challenges, several directions may be further explored in the future.

ACKNOWLEDGMENT

We are grateful to the authors of [47], [84], [71], [61], [68], [121], [116], [114], [96], [92], [100] for kindly releasing their experimental results or codes, as well as to the three anonymous reviewers for their constructive criticism, which has significantly improved our manuscript. Moreover, we thank Qiqi Bao for helpful discussions.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the International Conference on Machine Learning*, 2008, pp. 160–167.

[5] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 372–386.

[6] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1865–1873.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[9] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[10] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[11] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[12] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos, "Softcuts: a soft edge smoothness prior for color image super-resolution," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 969–981, 2009.

[13] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[14] Q. Yan, Y. Xu, X. Yang, and T. Q. Nguyen, "Single image superresolution based on gradient profile sharpness," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3187–3202, 2015.

[15] A. Marquina and S. J. Osher, "Image super-resolution by TV-regularization and Bregman iteration," *Journal of Scientific Computing*, vol. 37, no. 3, pp. 367–382, 2008.

[16] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[17] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 275–282.

[18] M. Aharon, M. Elad, A. Bruckstein *et al.*, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, p. 4311, 2006.

[19] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[20] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proceedings of the International Conference on Curves and Surfaces*, 2010, pp. 711–730.

[21] R. Timofte, V. De, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE international Conference on Computer Vision*, 2013, pp. 1920–1927.

[22] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proceedings of the Asian Conference on Computer Vision*, 2014, pp. 111–126.

[23] F. Cao, M. Cai, Y. Tan, and J. Zhao, "Image super-resolution via adaptive ℓ_p $(0 < p < 1)$ regularization and sparse representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 7, pp. 1550–1561, 2016.

[24] J. Liu, W. Yang, X. Zhang, and Z. Guo, "Retrieval compensated group structured sparsity for image super-resolution," *IEEE Transactions on Multimedia*, vol. 19, no. 2, pp. 302–316, 2017.

[25] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3791–3799.

[26] K. Zhang, D. Tao, X. Gao, X. Li, and J. Li, "Coarse-to-fine learning for single-image super-resolution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1109–1122, 2017.

[27] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, "A unified learning framework for single image super-resolution," *IEEE Transactions on Neural Networks and Learning systems*, vol. 25, no. 4, pp. 780–792, 2014.

[28] C. Deng, J. Xu, K. Zhang, D. Tao, X. Gao, and X. Li, "Similarity constraints-based structured output regression machine: An approach to image super-resolution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2472–2485, 2016.

[29] W. Yang, Y. Tian, F. Zhou, Q. Liao, H. Chen, and C. Zheng, "Consistent coding scheme for single-image super-resolution via independent dictionaries," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 313–325, 2016.

[30] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[31] H. A. Song and S.-Y. Lee, "Hierarchical representation using NMF," in *Proceedings of the International Conference on Neural Information Processing*, 2013, pp. 466–473.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[33] N. Rochester, J. Holland, L. Haibt, and W. Duda, "Tests on a cell assembly theory of the action of the brain, using a large digital computer," *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 80–93, 1956.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[35] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[36] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[37] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[38] J. F. Kolen and S. C. Kremer, *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*. IEEE, 2001. [Online]. Available: https://ieeexplore.ieee.org/document/5264952

[39] G. E. Hinton, "Learning multiple layers of representation," *Trends in Cognitive Sciences*, vol. 11, no. 10, pp. 428–434, 2007.

[40] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2011, pp. 1237–1242.

[41] D. CireşAn, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[42] R. Salakhutdinov and H. Larochelle, "Efficient learning of deep Boltzmann machines," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010, pp. 693–700.

[43] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[44] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[46] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[47] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 184–199.

[48] ——, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[49] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[50] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2018–2025.

[51] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[52] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 818–833.

[53] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, 2015, pp. 3431–3440.

[54] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[55] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, and Z. Wang, "Is the deconvolution layer the same as a convolutional layer?" *arXiv preprint arXiv:1609.07009*, 2016.

[56] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 391–407.

[57] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, "Accurate blur models vs. image priors in single image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2832–2839.

[58] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 675–678.

[59] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "TensorFlow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[60] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 2924–2932.

[61] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[63] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[65] ——, "Identity mappings in deep residual networks," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 630–645.

[66] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 550–558.

[67] D. Balduzzi, M. Frean, L. Leary, J. Lewis, K. W.-D. Ma, and B. McWilliams, "The shattered gradients problem: If resnets are the answer, then what is the question?" in *Proceedings of the International Conference on Machine Learning*, 2017, pp. 342–350.

[68] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[69] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, 2015, pp. 448–456.

[70] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3147–3155.

[71] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.

[72] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Association for the Advancement of Artificial Intelligence*, 2017, pp. 4278–4284.

[73] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[74] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2017, pp. 4470–4478.

[75] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4809–4817.

[76] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4539–4547.

[77] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[78] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.

[79] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 370–378.

[80] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the International Conference on International Conference on Machine Learning*, 2010, pp. 399–406.

[81] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, "Robust single image super-resolution via deep networks with sparse prior," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3194–3207, 2016.

[82] D. Liu, Z. Wang, N. Nasrabadi, and T. Huang, "Learning a mixture of deep networks for single image super-resolution," in *Proceedings of the Asian Conference on Computer Vision*, 2016, pp. 145–156.

[83] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, and S. Yan, "Deep edge guided recurrent residual learning for image super-resolution," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5895–5907, 2017.

[84] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 624–632.

[85] R. Dahl, M. Norouzi, and J. Shlens, "Pixel recursive super resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5439–5448.

[86] A. Singh and N. Ahuja, "Super-resolution using sub-band self-similarity," in *Proceedings of the Asian Conference on Computer Vision*, 2014, pp. 552–568.

[87] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proceedings of the International Conference on International Conference on Machine Learning*, 2016, pp. 1747–1756.

[88] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, "Conditional image generation with PixelCNN decoders," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 4790–4798.

[89] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical models and image processing*, vol. 53, no. 3, pp. 231–239, 1991.

[90] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep backprojection networks for super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1664–1673.

[91] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 606–615.

[92] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3262–3271.

[93] R. Timofte, V. De Smet, and L. Van Gool, "Semantic super-resolution: When and where is it useful?" *Computer Vision and Image Understanding*, vol. 142, pp. 1–12, 2016.

[94] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proceedings of the IEEE Global Conference on Signal and Information Processing*, 2013, pp. 945–948.

[95] T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1781–1790.

[96] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3929–3938.

[97] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1220–1234, 2019.

[98] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[99] K. Zhang, X. Gao, D. Tao, and X. Li, "Single image super-resolution with multiscale similarity learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1648–1659, 2013.

[100] A. Shocher, N. Cohen, and M. Irani, "zero-shot super-resolution using deep internal learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3118–3126.

[101] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2011, pp. 977–984.

[102] T. Michaeli and M. Irani, "Nonparametric blind super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 945–952.

[103] T. Tirer and R. Giryes, "Super-resolution based on image-adapted CNN denoisers: Incorporating generalization of training data and internal learning in test time," *arXiv preprint arXiv:1811.12866*, 2018.

[104] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[105] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 252–268.

[106] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2001, pp. 416–423.

[107] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 248–255.

[108] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135.

[109] Z. Yang, K. Zhang, Y. Liang, and J. Wang, "Single image super-resolution with a parameter economic residual-like convolutional neural network," in *Proceedings of the International Conference on Multimedia Modeling*, 2017, pp. 353–364.

[110] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 723–731.

[111] X. Fan, Y. Yang, C. Deng, J. Xu, and X. Gao, "Compressed multi-scale feature fusion network for single image super-resolution," *Signal Processing*, vol. 146, pp. 50–60, 2018.

[112] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for neural networks for image processing," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–51, 2017.

[113] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," *arXiv preprint arXiv:1511.05666*, 2015.

[114] J. Johnson, A. Alahi, and F.-F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 694–711.

[115] R. Mechrez, I. Talmi, F. Shama, and L. Zelnik-Manor, "Learning to maintain natural image statistics," *arXiv preprint arXiv:1803.04626*, 2018.

[116] R. Mechrez, I. Talmi, and L. Zelnik-Manor, "The contextual loss for image transformation with non-aligned data," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 768–783.

[117] K. Li and J. Malik, "Implicit maximum likelihood estimation," *arXiv preprint arXiv:1809.09087*, 2018.

[118] K. Li, S. Peng, and J. Malik, "Super-resolution via conditional implicit maximum likelihood estimation," *arXiv preprint arXiv:1810.01406*, 2018.

[119] F. Huszár, "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" *arXiv preprint arXiv:1511.05101*, 2015.

[120] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015.

[121] M. S. Sajjadi, B. Schölkopf, and M. Hirsch, "EnhanceNet: Single image super-resolution through automated texture synthesis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4501–4510.

[122] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[123] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 814–823.

[124] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the International Conference on Machine Learning*, 2017, pp. 214–223.

[125] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 271–279.

[126] D. J. Sutherland, H.-Y. Tung, H. Strathmann, S. De, A. Ramdas, A. Smola, and A. Gretton, "Generative models and model criticism via optimized maximum mean discrepancy," *arXiv preprint arXiv:1611.04488*, 2016.

[127] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6228–6237.